

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2000-299864

(P2000-299864A)

(43)公開日 平成12年10月24日 (2000.10.24)

(51)Int.Cl.<sup>7</sup>

H 0 4 N 7/32

識別記号

F I

H 0 4 N 7/137

テーマコード(参考)

Z 5 C 0 5 9

審査請求 未請求 請求項の数20 O L (全 13 頁)

(21)出願番号 特願平11-104088

(22)出願日 平成11年4月12日 (1999.4.12)

(71)出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 中山 忠義

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

(74)代理人 100076428

弁理士 大塚 康徳 (外2名)

Fターム(参考) 5C059 KK03 KK15 MA00 MA04 MA23

NN01 NN28 NN30 PP04 SS06

SS20 SS26 TA61 TB07 TC12

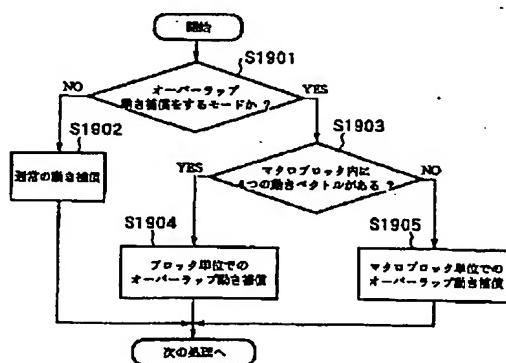
TC42 TC43 TD05 UA02 UA33

(54)【発明の名称】 動画像処理方法

(57)【要約】

【課題】 オーバーラップ動き補償予測符号化処理は、加算演算の回数が多いために処理時間がかかるてしまう。

【解決手段】 マクロブロック内の動きベクトルが同一であれば、ステップS1905においてマクロブロック単位でのオーバーラップ動き補償を行なうが、その際に、複数の重み係数マトリクスにおいて有効係数のみの演算を行なえば良いため、実質的な演算量を削減することができ、処理速度が向上する。



**【特許請求の範囲】**

**【請求項1】** 着目するブロック及びそれに隣接するブロックにおける動きベクトルのそれぞれに基づく第1の重み係数マトリクスより、複数ブロックからなるマクロブロックに対応する第2の重み係数マトリクスを生成し、

該第2の重み係数マトリクスを用いてマクロブロック単位でオーバーラップ動き補償予測符号化を行なうことを特徴とする動画像処理方法。

**【請求項2】** 前記オーバーラップ動き補償予測符号化の際に、前記第2の重み係数マトリクスに基づいて前記マクロブロックの予測画像を導出することを特徴とする請求項1記載の動画像処理方法。

**【請求項3】** 前記第1の重み係数マトリクスは前記ブロックのサイズに対応し、前記第2の重み係数マトリクスは前記マクロブロックのサイズに対応することを特徴とする請求項1記載の動画像処理方法。

**【請求項4】** 前記マクロブロックは、4つの前記ブロックよりなることを特徴とする請求項3記載の動画像処理方法。

**【請求項5】** 前記第2の重み係数マトリクスは、複数の前記第1の重み係数マトリクスを加算し、前記マクロブロックのサイズに連結することによって生成されることを特徴とする請求項3記載の動画像処理方法。

**【請求項6】** 前記隣接するブロックは、前記着目するブロックの上下左右に隣接するブロックであることを特徴とする請求項1記載の動画像処理方法。

**【請求項7】** 前記第2の重み係数マトリクスは、着目するマクロブロック及びその上、左、右に隣接するマクロブロックにおける動きベクトルのそれぞれに対応して複数生成されることを特徴とする請求項1記載の動画像処理方法。

**【請求項8】** 前記オーバーラップ動き補償予測符号化の際に、前記第2の重み係数マトリクスの有為係数に関してのみ重み付け加算演算が行われることを特徴とする請求項1記載の動画像処理方法。

**【請求項9】** 前記有為係数は、0又は8以外の係数であることを特徴とする請求項8記載の動画像処理方法。

**【請求項10】** 前記第2の重み係数マトリクスは、前記有為係数のみが保持されることを特徴とする請求項8記載の動画像処理方法。

**【請求項11】** 前記着目するマクロブロックは、1つの動きベクトルを有することを特徴とする請求項1記載の動画像処理方法。

**【請求項12】** 前記オーバーラップ動き補償予測符号化の際に、前記着目するマクロブロックの所定領域毎に、前記第2の重み係数マトリクスに基づく演算を施すことを特徴とする請求項11記載の動画像処理方法。

**【請求項13】** 前記隣接するマクロブロックの全てがそれぞれ1つの動きベクトルを有するか否かに応じて、

前記所定領域が異なることを特徴とする請求項12記載の動画像処理方法。

**【請求項14】** 前記隣接するマクロブロックの少なくとも1つが複数の動きベクトルを有する場合、前記所定領域は前記ブロックのサイズに対応することを特徴とする請求項13記載の動画像処理方法。

**【請求項15】** 更に、前記着目するマクロブロックが2つ以上の動きベクトルを有するか否かを判定し、該判定結果が否である場合に、前記オーバーラップ動き補償予測符号化を行なうことを特徴とする請求項11記載の動画像処理方法。

**【請求項16】** 前記着目するマクロブロックが2つ以上の動きベクトルを有すると判定された場合に、前記第1の重み係数マトリクスを用いて前記ブロック単位によるオーバーラップ動き補償予測符号化を行なうことを特徴とする請求項15記載の動画像処理方法。

**【請求項17】** 前記請求項1乃至16のいずれかに記載の動画像処理方法により符号化された動画像を復号する動画像処理方法。

**【請求項18】** 前記請求項1乃至16のいずれかに記載の動画像処理方法により動画像を符号化する動画像処理装置。

**【請求項19】** 前記請求項1乃至16のいずれかに記載の動画像処理方法により符号化された動画像を復号する動画像処理装置。

**【請求項20】** オーバーラップ動き補償予測符号化を行なう動画像処理方法のプログラムコードが記録された記録媒体であって、該プログラムは、着目するブロック及びそれに隣接するブロックにおける動きベクトルのそれぞれに基づく第1の重み係数マトリクスより、複数ブロックからなるマクロブロックに対応する第2の重み係数マトリクスを生成するコードと、該第2の重み係数マトリクスを用いてマクロブロック単位でオーバーラップ動き補償予測符号化を行なうコードと、を含むことを特徴とする記録媒体。

**【発明の詳細な説明】****【0001】**

**【発明の属する技術分野】** 本発明は動画像の動き補償予測符号化を行なう動画像処理方法に関する。

**【0002】**

**【従来の技術】** 近年、画像信号の符号化技術や該符号化データを通信したり蓄積したりする技術、また、該符号化データを安価なコストで復号するための半導体技術、復号した画像を表示する技術等、いわゆる画像情報を扱うインフラ技術の発達がめざましい。

**【0003】** これにより、動画像のような大量の情報を有する画像であっても遠隔地へ転送することができ、多地点間におけるより高度な情報交換が可能になってきた。

**【0004】** 動画像の符号化においては、フレーム間の

画像信号の相関性を利用して情報量を大幅に削減することが可能である。この特性を利用した代表的な符号化技術として、所謂動き補償予測符号化方式が知られている。該符号化方式においては、所定の画素ブロック毎に、参照すべき画像の相対位置を表わす動きベクトルと称する情報を参照する。そして、該動きベクトル情報に基づいて得られる画素情報を予測値とし、該予測値と実際の画素値との差分に対してDCT等の直交変換を施すことにより、該変換情報を人間の視覚特性に基づいて削減することができる。

【0005】このようなブロック単位での符号化方式において、圧縮率を上げるために変換情報を大幅に削減すると、ブロック状の歪が出現しやすくなってしまう。一般に符号の転送レートが低い装置において使用する圧縮方式においては、圧縮率を上げることは必須であるため、ブロック状の歪は避け難いものとなっていた。

【0006】そこで、ITU-T勧告H.263規格やMPEG4規格においては、前記ブロック状の歪を軽減するため、オーバーラップ動き補償という処理を行なっている。

【0007】一般的な動き補償予測においては、フレーム画像内の着目ブロックを、参照画像上で動きベクトル分だけずらしたものと該着目ブロックの予測画像とすることにより、符号化を行なう。この場合、隣接ブロック間で動きベクトルが同一であれば、該隣接する2つのブロックの予測画像データは隣接部において連続性があるため、予測画像データ上ではブロック歪が生じることは無い。ところが、隣接ブロック間で動きベクトルに若干の差がある場合、その動きベクトルの差が、隣接する2つのブロックの予測画像の隣接部における不連続性となって現れる。

【0008】そこで、着目ブロックの動きベクトルのみを用いて予測を行なう代わりに、隣接するブロックの動きベクトルを反映するような予測を行なうことによって、上記不連続性を軽減することが考えられる。

【0009】オーバーラップ動き補償では、着目ブロックの動きベクトルを用いて予測した画像データ(1ブロック分)、隣接する4つのブロックの動きベクトルを用いて予測した画像データ(1ブロック分×4)の各々に重み係数を乗じ、それらを合算したデータを最終的な予測値(画像)とする。

【0010】

【発明が解決しようとする課題】しかしながら、従来のオーバーラップ動き補償においては、加算演算の回数が多く、処理時間がかかるといった問題があった。

【0011】本発明は上記問題を解決するためになされたものであり、動きベクトル予測に伴って生じる隣接ブロック間の不連続性の歪を軽減するオーバーラップ動き補償を高速に可能とする動画像処理方法を提供することを目的とする。

### 【0012】

【課題を解決するための手段】上記目的を達成するための一手法として、本発明の動画像処理方法は以下の工程を備える。

【0013】即ち、着目するブロック及びそれに隣接するブロックにおける動きベクトルのそれぞれに基づく第1の重み係数マトリクスより、複数ブロックからなるマクロブロックに対応する第2の重み係数マトリクスを生成し、該第2の重み係数マトリクスを用いてマクロブロック単位でオーバーラップ動き補償予測符号化を行なうこととする。

### 【0014】

【発明の実施の形態】以下、本発明に係る一実施形態について詳細に説明する。

【0015】<第1実施形態>まず、一般的なオーバーラップ動き補償について簡単に説明する。

【0016】図1において、符号化または復号対象であるフレーム画像101内の座標(x0, y0)を位置情報とする着目ブロック102を、参照画像104上で動きベクトル103分だけずらした位置にある1ブロックの画像データを、着目ブロック102の予測画像105とする。これが、一般的な動き補償による予測である。

【0017】このような動き補償予測において、隣接ブロック間で動きベクトルが同一であれば、該隣接する2つのブロックの予測画像データは隣接部において連続性があるため、予測画像データ上ではブロック歪が生じることは無い。ところが、隣接ブロック間で動きベクトルに若干の差がある場合、その動きベクトルの差が、隣接する2つのブロックの予測画像の隣接部における不連続性となって現れる。

【0018】そこで、着目ブロックの動きベクトルのみを用いて予測を行なう代わりに、隣接するブロックの動きベクトルを反映するような予測を行なうことによって、上記不連続性を軽減することが考えられる。

【0019】オーバーラップ動き補償では、着目ブロックの動きベクトルを用いて予測した画像データ(1ブロック分)、隣接する4つのブロックの動きベクトルを用いて予測した画像データ(1ブロック分×4)の各々に重み係数を乗じ、それらを合算したデータを最終的な予測値(画像)とする。

【0020】具体的には、着目ブロックの動きベクトルを用いて予測した画像データに対しては図2に示す重み係数マトリクスを乗じ、また、着目ブロックの上、下、左、右のそれぞれに隣接するブロックの動きベクトルを用いて予測した画像データに対してはそれぞれ図3、図4、図5、図6に示す重み係数マトリクスを乗じる。そして、それら5画像分を全て足し合わせた後に更に4を加算し、8で割った値を予測値(画像)とする。

【0021】上記演算は即ち、以下に示す(1)式で表される。

## 【0022】

$$p(x, y) = (q(x, y)H2(i, j) + u(x, y)H3(i, j) + d(x, y)H4(i, j) + l(x, y)H5(i, j) + r(x, y)H6(i, j) + 4)/8 \quad \dots (1)$$

ここで、(x, y)は画像全体における着目ブロック中の各画素位置を表わす座標であり、i, jは着目ブロックの左上の座標を(x0, y0)とした時、i=x-x0, j=y-y0という値をとる。また、q(x, y)は着目ブロックの動きベクトルを用いて予測した画像データであり、u(x, y), d(x, y), l(x, y), r(x, y)はそれぞれ、着目ブロックの上、下、左、右に隣接するブロックの動きベクトルを用いて予測した画像データである。これらの画像データq(x, y), u(x, y), d(x, y), l(x, y), r(x, y)は、それぞれの動きベクトルを(MV1x, MV1y), (MV2x, MV2y), (MV3x, MV3y), (MV4x, MV4y), (MV5x, MV5y)とし、予測の基になる参照画像をref(i, j)とすると、以下の様に表される。

【0023】 $q(x, y) = \text{ref}(x+MV1x, y+MV1y)$

$u(x, y) = \text{ref}(x+MV2x, y+MV2y)$

$d(x, y) = \text{ref}(x+MV3x, y+MV3y)$

$l(x, y) = \text{ref}(x+MV4x, y+MV4y)$

$r(x, y) = \text{ref}(x+MV5x, y+MV5y)$

また、H2(i, j), H3(i, j), H4(i, j), H5(i, j), H6(i, j)は、各々図2、図3、図4、図5、図6に示す重み係数マトリクスであり、それぞれ $0 \leq i, j \leq 7$ の範囲においてのみ、値が定義されている。

【0024】以下、本実施形態におけるオーバラップ動き補償について説明する。本実施形態においては、マクロブロック単位によるオーバラップ動き補償を行なうこととする特徴とする。

【0025】ITU-T勧告H.263規格やMPEG4規格において

$$\begin{aligned} p(x, y) &= (q(x, y)H2(i, j) + u(x, y)H3(i, j) + q(x, y)H4(i, j) \\ &\quad + l(x, y)H5(i, j) + q(x, y)H6(i, j) + 4)/8 \\ &= (q(x, y)(H2(i, j) + H4(i, j) + H6(i, j)) \\ &\quad + u(x, y)H3(i, j) + l(x, y)H5(i, j) + 4)/8 \end{aligned} \quad \dots (2)$$

同様にブロックbについては、(1)式においてd(x, y)とl(x, y)をq(x, y)に置き換えることができるため、以下の

$$\begin{aligned} p(x, y) &= (q(x, y)H2(i, j) + u(x, y)H3(i, j) + q(x, y)H4(i, j) \\ &\quad + q(x, y)H5(i, j) + r(x, y)H6(i, j) + 4)/8 \\ &= (q(x, y)(H2(i, j) + H4(i, j) + H5(i, j)) \\ &\quad + u(x, y)H3(i, j) + r(x, y)H6(i, j) + 4)/8 \end{aligned} \quad \dots (3)$$

またブロックcについては、上に隣接するブロックaと右に隣接するブロックdが同一マクロブロック内にあるため、ブロックcと同じ動きベクトルを持つ。さらに下に隣接するブロックは符号化もしくは復号化処理がまだ行われておらず、動きベクトル情報が得られないため、ブ

$$\begin{aligned} p(x, y) &= (q(x, y)H2(i, j) + q(x, y)H3(i, j) + q(x, y)H4(i, j) \\ &\quad + l(x, y)H5(i, j) + q(x, y)H6(i, j) + 4)/8 \\ &= (q(x, y)(H2(i, j) + H3(i, j) + H4(i, j) + H6(i, j)) \\ &\quad + l(x, y)H5(i, j) + 4)/8 \end{aligned} \quad \dots (4)$$

同様にブロックdについては、u(x, y), d(x, y), l(x, y)をq(x, y)に置き換えることができるため、以下の(5)式

は、 $8 \times 8$ のブロックという単位の上位に $16 \times 16$ のマクロブロックという単位があり、該マクロブロックは図7に示すように4つのブロックより成る。該マクロブロックにおいて、各ブロックがそれぞれ別の動きベクトルを持つことを可能とするモードが存在するが、マクロブロック全体における動きの相関が強い場合には、情報量削減のために該モードにおいても該マクロブロックを1つの動きベクトルで代表する場合がある。

【0026】以下、マクロブロック全体で1つの動きベクトルしか持たない、即ち、マクロブロックを構成する各ブロックの動きベクトルが等しい場合について考える。

【0027】上述した一般的なオーバーラップ動き補償予測を示す(1)式を、図7に示すマクロブロックを構成する4つのブロックa, b, c, dにそれぞれ適用すると、該(1)式は各ブロック毎に異なる演算式に変形される。

【0028】まず、ブロックaについて考える。ブロックaが着目ブロックであれば、ブロックbとブロックcはそれぞれ右と下に隣接するブロックであり、この場合マクロブロック内で1つの動きベクトルしか持たないため、ブロックbとブロックcはブロックaと同一の動きベクトルを持つことになる。従って、(1)式においてd(x, y)とr(x, y)をq(x, y)で置き換えることができる。よって、ブロックaに対するオーバーラップ動き補償予測については、以下の(2)式で示される演算式を得る。

【0029】

$$\begin{aligned} p(x, y) &= (q(x, y)H2(i, j) + u(x, y)H3(i, j) + q(x, y)H4(i, j) \\ &\quad + q(x, y)H5(i, j) + q(x, y)H6(i, j) + 4)/8 \\ &= (q(x, y)(H2(i, j) + H4(i, j) + H6(i, j)) \\ &\quad + u(x, y)H3(i, j) + q(x, y)H5(i, j) + 4)/8 \end{aligned} \quad \dots (3)$$

(3)式で示される演算式を得る。

【0030】

$$\begin{aligned} p(x, y) &= (q(x, y)H2(i, j) + u(x, y)H3(i, j) + q(x, y)H4(i, j) \\ &\quad + q(x, y)H5(i, j) + r(x, y)H6(i, j) + 4)/8 \\ &= (q(x, y)(H2(i, j) + H4(i, j) + H5(i, j)) \\ &\quad + u(x, y)H3(i, j) + r(x, y)H6(i, j) + 4)/8 \end{aligned} \quad \dots (3)$$

ブロックcの動きベクトルで置き換えることができる。よって、前記(1)式において、u(x, y), d(x, y), r(x, y)をq(x, y)に置き換えることができ、以下の(4)式で示される演算式を得る。

【0031】

$$\begin{aligned} p(x, y) &= (q(x, y)H2(i, j) + q(x, y)H3(i, j) + q(x, y)H4(i, j) \\ &\quad + l(x, y)H5(i, j) + q(x, y)H6(i, j) + 4)/8 \\ &= (q(x, y)(H2(i, j) + H3(i, j) + H4(i, j) + H6(i, j)) \\ &\quad + l(x, y)H5(i, j) + 4)/8 \end{aligned} \quad \dots (4)$$

で示される演算式を得る。

【0032】

$$\begin{aligned}
 p(x, y) &= (q(x, y)H2(i, j) + q(x, y)H3(i, j) + q(x, y)H4(i, j) \\
 &\quad + q(x, y)H5(i, j) + r(x, y)H6(i, j) + 4) / 8 \\
 &= (q(x, y)(H2(i, j) + H3(i, j) + H4(i, j) + H5(i, j)) \\
 &\quad + r(x, y)H6(i, j) + 4) / 8
 \end{aligned} \tag{5}$$

以上の(2)～(5)式によれば、4つのブロックa, b, c, dにおける予測画像データq(x, y)に対する重み係数マトリクスは、それぞれ

$$\begin{aligned}
 H2(i, j) + H4(i, j) + H6(i, j) \\
 H2(i, j) + H4(i, j) + H5(i, j) \\
 H2(i, j) + H3(i, j) + H4(i, j) + H6(i, j) \\
 H2(i, j) + H3(i, j) + H4(i, j) + H5(i, j)
 \end{aligned}$$

となり、これらをそれぞれ図8(a)～(d)に示す。

【0033】該4つのブロックa, b, c, dをマクロブロック単位で処理しようとすると、重み係数マトリクスもマクロブロックに対応した大きさ、すなわち16×16サイズとする必要がある。該重み係数マトリクスとしては、図8(a)～(d)に示した4つの重み係数マトリクスを図9に示すように連結することで得られる。ここで、このマトリクスをM2(i, j)と定義する。

【0034】また、上に隣接するブロックの動きベクトルを用いて予測した画像データu(x, y)に対応する16×16の重み係数マトリクスは、図3に示すH3(i, j)を2つ横に連結し、下半分である16×8の領域には0を埋め込んだも

$$\begin{aligned}
 p(x, y) &= (q(x, y)M2(i, j) + u(x, y)M3(i, j) + l(x, y)M5(i, j) \\
 &\quad + r(x, y)M6(i, j) + 4) / 8
 \end{aligned} \tag{6}$$

(6)式によれば、上記(1)式において0～7の範囲であったi, jの値が、0～15まで広がる。ここで(6)式によれば、図9に示すM2(i, j)の3/8を占める、係数が8である要素については、p(x, y)=q(x, y)となるので、計算を行なう必要がなくなる。即ち、実際に演算処理を実行しなければならない画素が、全体の5/8まで減少することが分かる。

【0040】このように、本実施形態のマクロブロック単位によるオーバーラップ動き補償により、マクロブロックに対応した重み係数マトリクスにおいて有為係数のみの演算を行なえば良いため、一般的なブロック単位によるオーバーラップ動き補償と比べて演算量を実質的に削減することができる。

【0041】以上の説明においては、着目マクロブロックは1つの動きベクトルしか持っていないという前提に基づき、新たなオーバーラップ動き補償の演算式や重み係数マトリクスを導出してきた。しかしながら実際には、上または左右に隣接する3つのマクロブロックも各々1つの動きベクトルしか持たない場合や、各々が4つの動きベクトルを持つ場合、あるいは、各々のブロックによって動きベクトル数が異なる場合などが考えられる。

【0042】従って本実施形態のオーバーラップ動き補償においては、上または左右に隣接する3つのマクロブロックの各々が、1つの動きベクトルしかない場合とそうでない場合とのそれぞれについて、異なる処理を行な

るので表される。これをM3(i, j)と定義し、図10に示す。

【0035】また、左に隣接するブロックの動きベクトルを用いて予測した画像データl(x, y)に対応する16×16の重み係数マトリクスは、図5に示すH5(i, j)を2つ縦に連結し、右半分である8×16の領域には0を埋め込んだもので表される。これをM5(i, j)と定義し、図11に示す。

【0036】また、右に隣接するブロックの動きベクトルを用いて予測した画像データr(x, y)に対応する16×16の重み係数マトリクスは、M5(i, j)を左右反転したもので表される。これをM6(i, j)と定義し、図12に示す。

【0037】尚、下に隣接するブロックの動きベクトルを用いて予測した画像データd(x, y)に対応する重み係数マトリクスは、すでにM2(i, j)に反映済みである。

【0038】これら重み係数マトリクスM2(i, j), M3(i, j), M5(i, j), M6(i, j)を用いて、マクロブロック単位のオーバーラップ動き補償の演算式を記述すると、(6)式のようになる。

#### 【0039】

うことを特徴とする。

【0043】図13に、本実施形態における動き補償処理の概要フローチャートを示す。まずステップS1901において、オーバーラップ動き補償を行なうモードであるか否かを識別し、該モードでなければ、S1902において通常の動き補償を実行する。ここでは、重み付け演算は行なわず、

$$p(x, y) = q(x, y) = \text{ref}(x + MV1x, y + MV1y)$$

により予測画像を生成する。

【0044】一方、オーバーラップ動き補償を行なうモードであると識別された場合には、ステップS1903において、マクロブロック内に4つの動きベクトルがあるか否かを識別する。4つの動きベクトルがあれば、ステップS1904において(1)式に示した一般的なブロック単位によるオーバーラップ動き補償処理を行なう。一方、動きベクトルが4つではなく、1つしか無い場合には、ステップS1905において本実施形態の特徴であるマクロブロック単位でのオーバーラップ動き補償処理を行なう。

【0045】図14に、上記ステップS1905における本実施形態のマクロブロック単位によるオーバーラップ動き補償処理の詳細フローチャートを示す。

【0046】まずステップS2001において、上、左、右にそれぞれ隣接するマクロブロックにおける動きベクトルの数が、各々1つしか無いか否かを判別し、各々1つしか無い場合には、ステップS2002～S2005の処理を行な

う。

【0047】まずステップS2002においては、 $16 \times 16$ の着目マクロブロックのうち、図15に示す領域21aについてのみ、

$p(x, y) = M3(i, j) u(x, y) = M3(i, j) \text{ref}(x+MV2x, y+MV2y)$   
なる重み付け演算を行なう。次にステップS2003において、同じく図16に示す領域22aについてのみ、  
 $p(x, y) = p(x, y) + M5(i, j) l(x, y) = p(x, y) + M5(i, j) \text{ref}(x+MV4x, y+MV4y)$

なる重み付け演算を行なう。尚、該式においては右辺の  $p(x, y)$  に対する加算処理により、左辺の  $p(x, y)$  が新たに得られることを示す。そしてステップS2004において、図16に示す領域22bについてのみ、

$p(x, y) = p(x, y) + M6(i, j) l(x, y) = p(x, y) + M6(i, j) \text{ref}(x+MV5x, y+MV5y)$

なる重み付け演算を行なう。最後にステップS2005において、着目マクロブロックにおける図15に示す領域21a、図16に示す領域22a、22bに対して、

$p(x, y) = (p(x, y) + M2(i, j) q(x, y) + 4) / 8$   
 $= (p(x, y) + M2(i, j) \text{ref}(x+MV1x, y+MV1y) + 4) / 8$

を演算し、着目マクロブロックの残りの領域に対しては、

$p(x, y) = q(x, y) = \text{ref}(x+MV1x, y+MV1y)$   
のコピー処理を行なう。

【0048】上述したステップS2002～S2005の処理は、代入処理や加算処理を配慮すれば、その演算の順番を入れ替えることができ、例えばステップS2005の処理を最初に行なうことでも可能である。この場合、上記ステップS2005における

$p(x, y) = (p(x, y) + M2(i, j) q(x, y) + 4) / 8$   
 $= (p(x, y) + M2(i, j) \text{ref}(x+MV1x, y+MV1y) + 4) / 8$

なる加算処理は、

$p(x, y) = M2(i, j) q(x, y)$   
 $= M2(i, j) \text{ref}(x+MV1x, y+MV1y)$

なる代入処理のみとなり、ステップS2002～S2004の他の処理ブロックで加算処理を行なうことになる。もちろん、演算の最後には、4を加算して8で割るといった丸め処理を行なう必要がある。

【0049】一方、ステップS2001における判別結果が異なる場合、即ち、隣接する3つのマクロブロックの内、少なくとも1つのマクロブロックが動きベクトルを4つ持っている場合、ステップS2010～S2016の処理を行なう。

【0050】この場合、隣接する3つのマクロブロックをブロックに分割し、図17に示すように、着目するマクロブロックCとそれに隣接する6つのブロックU1, U2, L1, L2, R1, R2としてとらえる。

【0051】尚、同じマクロブロックに属する2つのブロックは、同一の動きベクトルを持つ（マクロブロック内に動きベクトルを1つしか持たない場合）こともあれ

ば、異なる動きベクトルを持つ（マクロブロック内に4つの動きベクトルを持つ場合）こともある。ここで、6つのブロックU1, U2, L1, L2, R1, R2に対応する動きベクトルをそれぞれ、(MVb1x, MVb1y), (MVb2x, MVb2y), (MVb3x, MVb3y), (MVb4x, MVb4y), (MVb5x, MVb5y), (MVb6x, MVb6y)とする。

【0052】まずステップS2010においては、 $16 \times 16$ の着目マクロブロックのうち、図18に示す領域24aについてのみ、

$p(x, y) = M3(i, j) u(x, y) = M3(i, j) \text{ref}(x+MVb1x, y+MVb1y)$   
なる重み付け演算を行なう。次にステップS2011において、同様に図18に示す領域24bについてのみ、  
 $p(x, y) = p(x, y) + M3(i, j) u(x, y) = p(x, y) + M3(i, j) \text{ref}(x+MVb2x, y+MVb2y)$

なる重み付け演算を行なう。尚、上述したように、右辺の  $p(x, y)$  に対する加算処理により、左辺の  $p(x, y)$  が新たに得られることを示す。そしてステップS2012において、図19に示す領域25aについてのみ、

$p(x, y) = p(x, y) + M5(i, j) l(x, y) = p(x, y) + M5(i, j) \text{ref}(x+MVb3x, y+MVb3y)$

なる重み付け演算を行なう。そしてステップS2013において、図19に示す領域25bについてのみ、

$p(x, y) = p(x, y) + M5(i, j) l(x, y) = p(x, y) + M5(i, j) \text{ref}(x+MVb4x, y+MVb4y)$

なる重み付け演算を行なう。次にステップS2014において、図19に示す領域25cについてのみ、

$p(x, y) = p(x, y) + M6(i, j) r(x, y) = p(x, y) + M6(i, j) \text{ref}(x+MVb5x, y+MVb5y)$

なる重み付け演算を行なう。次にステップS2015において、図19に示す領域25dについてのみ、

$p(x, y) = M6(i, j) r(x, y) = p(x, y) + M6(i, j) \text{ref}(x+MVb6x, y+MVb6y)$

なる重み付け演算を行なう。そして最後にステップS2016において、着目マクロブロックにおける図18に示す領域24a, 24b、及び図19に示す領域25a, 25b, 25c, 25dに対して、

$p(x, y) = (p(x, y) + M2(i, j) q(x, y) + 4) / 8$   
 $= (p(x, y) + M2(i, j) \text{ref}(x+MV1x, y+MV1y) + 4) / 8$

を演算し、着目マクロブロックの残りの領域には、

$p(x, y) = q(x, y) = \text{ref}(x+MV1x, y+MV1y)$   
のコピー処理を行なう。

【0053】尚、上述したステップS2002～S2005の場合と同様、ステップS2010～S2016の処理についても、その順番を入れ替えることが可能である。

【0054】以上説明したように本実施形態によれば、マクロブロック単位によるオーバーラップ動き補償を行なうことにより、アクセスする画素数を減少し、加算演算の回数を減らすことができる。従ってオーバーラップ動き補償処理を高速化することが可能となる。

【0055】<第2実施形態>以下、本発明に係る第2

実施形態について説明する。尚、第2実施形態において、着目マクロブロックに隣接するマクロブロックはそれぞれ1つの動きベクトルしか持たないものと仮定する。尚、4つの動きベクトルを持つマクロブロックが隣接する場合には、上述した第1実施形態と同様に場合分けを行ない、それぞれに応じた処理を実行すれば良い。

【0056】第2実施形態においては、上述した第1実施形態で示したマクロブロック単位によるオーバーラップ動き補償処理を、画像の上部境界のマクロブロックに適用したことを特徴とする。

【0057】画像の上部境界に位置する着目ブロックにおいては、上に隣接するブロックが存在しないために、該上に隣接するブロックに対応する動きベクトルは、着目ブロックの動きベクトルで置き換えて処理される。即ち、第1実施形態で示した(6)式において、 $u(i, j)$ を $q(i, j)$ に置き換えることにより、演算を行なう。

$$\begin{aligned} p(x, y) &= (q(x, y)M2(i, j) + q(x, y)M3(i, j) + l(x, y)M5(i, j) \\ &\quad + r(x, y)M6(i, j) + 4)/8 \\ &= (q(x, y)(M2(i, j) + M3(i, j)) + l(x, y)M5(i, j) \\ &\quad + r(x, y)M6(i, j) + 4)/8 \end{aligned} \quad \cdots(7)$$

ここで(7)式によれば、図20に示す $M2(i, j) + M3(i, j)$ の1/2を占める、係数が8である要素については、 $p(x, y) = q(x, y)$ となるので、計算を行なう必要がなくなる。即ち、実際に演算処理を実行しなければならない画素が、全体の1/2まで減少することが分かる。

【0061】以上説明したように本実施形態によれば、画像の上部境界にあるマクロブロックのオーバーラップ動き補償処理をさらに高速化することができる。

【0062】例えば、符号化対象である画像のフォーマットがテレビジョンの共通中間フォーマットであるQCIF(Quarter Common Intermediate Format)であった場合、画像全体のマクロブロックの数は $11 \times 9$ 個であり、そのうち11個のマクロブロックが画像の上部境界に相当する。従って、画像全体の1/9の領域において、第2実施形態による効果が得られる。

【0063】さらに、画像フォーマットがサブQCIFであった場合、画像全体のマクロブロックの数は $8 \times 6$ 個であり、そのうち8個のマクロブロックが画像の上部境界にあるため、画像全体の1/6の領域において、第2実施形態による効果が得られる。

【0064】<第3実施形態>以下、本発明に係る第3実施形態について説明する。尚、第3実施形態において、着目マクロブロックに隣接するマクロブロックはそれぞれ1つの動きベクトルしか持たないものと仮定する。尚、4つの動きベクトルを持つマクロブロックが隣接する場合には、上述した第1実施形態と同様に場合分

【0058】従って、図9に示す重み係数マトリクス $M2(i, j)$ と図10に示す重み係数マトリクス $M3(i, j)$ の加算結果である、図20に示す重み係数マトリクスを用いることにより、マクロブロック単位のオーバーラップ動き補償処理を行なうことができる。即ち、着目マクロブロックの動きベクトルを用いて予測した画像データに対しては図20に示す重み係数マトリクスを乗じ、左右に隣接するブロックの動きベクトルを用いて予測した画像データに対しては、第1実施形態において図11、図12に示した重み係数マトリクスを乗じる。そして、それらを足し合わせてさらに4を加えた後に、8で割った値を予測値(画像)とする。

【0059】上記演算を数式によって表現すると、下記に示す(7)式が得られる。

【0060】

けを行ない、それぞれに応じた処理を実行すれば良い。

【0065】第3実施形態においては、上述した第1実施形態で示したマクロブロック単位によるオーバーラップ動き補償処理を、画像の左境界及び右境界のマクロブロックに適用したことを特徴とする。

【0066】画像の左境界に位置する着目ブロックにおいては、左に隣接するブロックが存在せず、また、画像の右境界に位置する着目ブロックにおいては、右に隣接するブロックが存在しない。従って、これら存在しない隣接ブロックに対応する動きベクトルは、着目ブロックの動きベクトルで置き換えて処理される。即ち、第1実施形態で示した(6)式において、画像の左境界では $l(i, j)$ を $q(i, j)$ に置き換え、右境界では $r(i, j)$ を $q(i, j)$ に置き換えることにより、演算を行なう。

【0067】従って左境界においては、図9に示す重み係数マトリクス $M2(i, j)$ と図11に示す重み係数マトリクス $M5(i, j)$ の加算結果である、図21に示す重み係数マトリクスを用いて、また右境界においては、図9に示す重み係数マトリクス $M2(i, j)$ と図12に示す重み係数マトリクス $M6(i, j)$ の加算結果である、図22に示す重み係数マトリクスを用いることにより、マクロブロック単位のオーバーラップ動き補償処理を行なうことができる。

【0068】第3実施形態における左境界及び右境界のオーバーラップ動き補償の演算を数式によって表現すると、下記に示す(8)式及び(9)式が得られる。

【0069】

$$\begin{aligned} p(x, y) &= (q(x, y)M2(i, j) + u(x, y)M3(i, j) + q(x, y)M5(i, j) \\ &\quad + r(x, y)M6(i, j) + 4)/8 \\ &= (q(x, y)(M2(i, j) + M5(i, j)) + u(x, y)M3(i, j) \\ &\quad + r(x, y)M6(i, j) + 4)/8 \end{aligned} \quad \cdots(8)$$

$$\begin{aligned}
 p(x, y) &= (q(x, y)M2(i, j) + u(x, y)M3(i, j) + l(x, y)M5(i, j) \\
 &\quad + q(x, y)M6(i, j) + 4) / 8 \\
 &= (q(x, y)(M2(i, j) + M6(i, j)) + u(x, y)M3(i, j) \\
 &\quad + l(x, y)M5(i, j) + 4) / 8
 \end{aligned} \tag{9}$$

ここで、例えば(8)式によれば、図21に示すM2(i, j)+M5(i, j)の9/16を占める、係数が8である要素については、 $p(x, y) = q(x, y)$ となるので、計算を行なう必要がなくなる。即ち、実際に演算処理を実行しなければならない画素が、全体の7/16まで減少することが分かる。また(9)式においても、同様に処理すべき画素数が全体の7/16となる。

【0070】以上説明したように本実施形態によれば、画像の左境界及び右境界にあるマクロブロックのオーバーラップ動き補償処理をさらに高速化することができる。

【0071】例えば、画像のフォーマットがQCIFであった場合、画像全体の11×9個のマクロブロックのうち、9個が画像の左境界に、また9個が右境界に存在する。従って、画像全体の2/11の領域において、第3実施形態による効果が得られる。

【0072】さらに、画像フォーマットがサブQCIFであった場合、画像全体のマクロブロックの数は8×6個であり、そのうち12個のマクロブロックが画像の左又は右境界にあるため、画像全体の1/4の領域において、第3実施形態による効果が得られる。

【0073】<第4実施形態>以下、本発明に係る第4実施形態について説明する。

【0074】上述した第1乃至第3実施形態によれば、オーバーラップ動き補償の処理速度を向上させることができる。第4実施形態においては、上記各実施形態において使用される重み係数マトリクスの格納領域を節約することを特徴とする。

【0075】上記各実施形態においては、オーバーラップ動き補償をマクロブロック単位で行なうために各重み係数マトリクスは16×16のサイズとなっているが、該マトリクスにおいて、係数が0または8である部分については実際の処理を行なう必要がないため、このような係数は保持しておく必要がない。

【0076】例えば、図10、図11、図12に示す重み係数マトリクスについては、係数が0でない、16×4または4×16の有為係数部分のみを保持すれば良い。また、図9に示す重み係数マトリクスについては、係数が8でない、4×16の有為係数部分を2つと8×4の有為係数部分を1つ保持すれば良い。また図20に示す重み係数マトリクスについては、係数が8でない、4×16の有為係数部を2つ保持すれば良い。また、図21、図22に示す重み係数マトリクスについては、係数が8でない、16×4の有為係数部分を1つと4×12の有為係数部分を1つ保持すれば良い。

【0077】また、図9～図12、及び図20に示す重み係数マトリクスは上下又は左右に対象性があるため、さらにその半分の有為係数のみを保持すれば良い。

【0078】以上説明したように第4実施形態によれば、上述した第1乃至第3実施形態におけるオーバーラップ動き補償を行なう際に、参照する重み係数マトリクスの最低限必要な係数のみを保持することにより、そのメモリ容量を節約することができる。

【0079】<他の実施形態>なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダ、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0080】また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。

【0081】この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

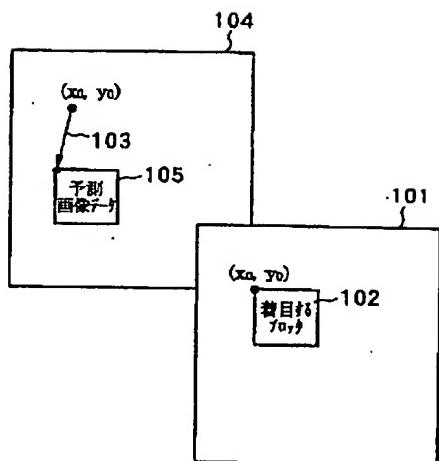
【0082】プログラムコードを供給するための記憶媒体としては、例えば、フロッピディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【0083】また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0084】さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明したフローチャートに対応するプログラムコードを格納することになる。



【図 1】



【図 5】

2	1	1	1	0	0	0	0	0	0	0	0	1	1	1	2
2	2	1	1	0	0	0	0	0	0	0	0	1	1	2	2
2	2	1	1	0	0	0	0	0	0	0	0	1	1	2	2
2	2	1	1	0	0	0	0	0	0	0	0	1	1	2	2
2	2	1	1	0	0	0	0	0	0	0	0	1	1	2	2
2	2	1	1	0	0	0	0	0	0	0	0	1	1	2	2
2	2	1	1	0	0	0	0	0	0	0	0	1	1	2	2
2	1	1	1	0	0	0	0	0	0	0	0	1	1	1	2

【図 6】

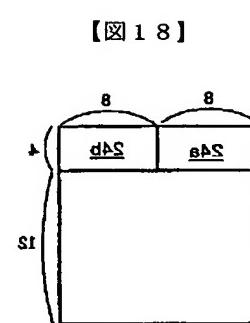
【図 8】

4	5	5	5	6	8	6	6
5	5	5	6	6	7	7	
5	5	6	6	7	7	7	
5	5	6	6	7	7	7	
6	6	7	7	8	8	8	8
6	6	7	7	8	8	8	8
6	6	7	7	8	8	8	8
6	6	7	7	8	8	8	8

(a)

6	6	6	6	5	5	5	4
7	7	6	6	5	5	5	
7	7	7	6	6	5	5	
7	7	7	6	6	5	5	
8	8	8	8	7	7	6	6
8	8	8	8	7	7	6	6
8	8	8	8	7	7	6	6
8	8	8	8	7	7	6	6

(b)



【図 18】

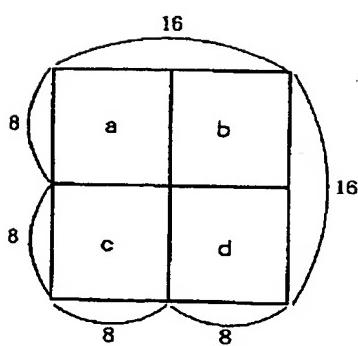
6	7	7	7	8	8	8	8
6	6	7	7	8	8	8	8
6	6	7	7	8	8	8	8
6	6	7	7	8	8	8	8
6	6	7	7	8	8	8	8
6	6	7	7	8	8	8	8
6	6	7	7	8	8	8	8
6	7	7	7	8	8	8	8

(c)

8	8	8	8	7	7	7	6
8	8	8	8	7	7	6	
8	8	8	8	7	7	6	
8	8	8	8	7	7	6	
8	8	8	8	7	7	6	
8	8	8	8	7	7	6	
8	8	8	8	7	7	6	
8	8	8	8	7	7	6	

(d)

【図 7】



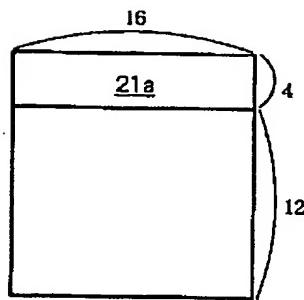
〔図9〕

[図 10]

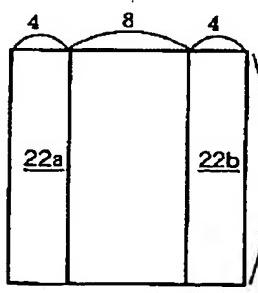
【图11】

[図12]

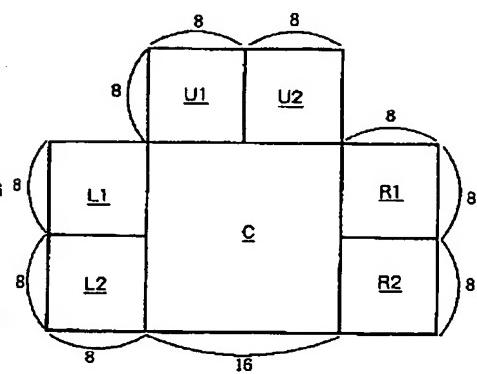
[図15]



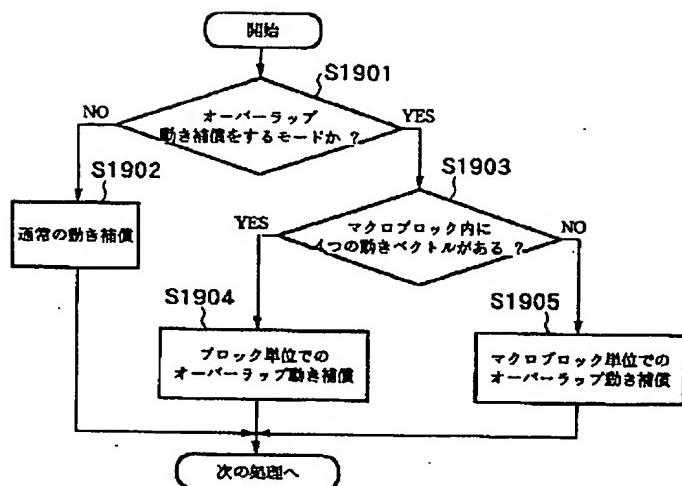
【図16】



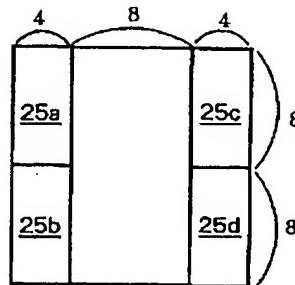
〔図17〕



〔图 1-3〕



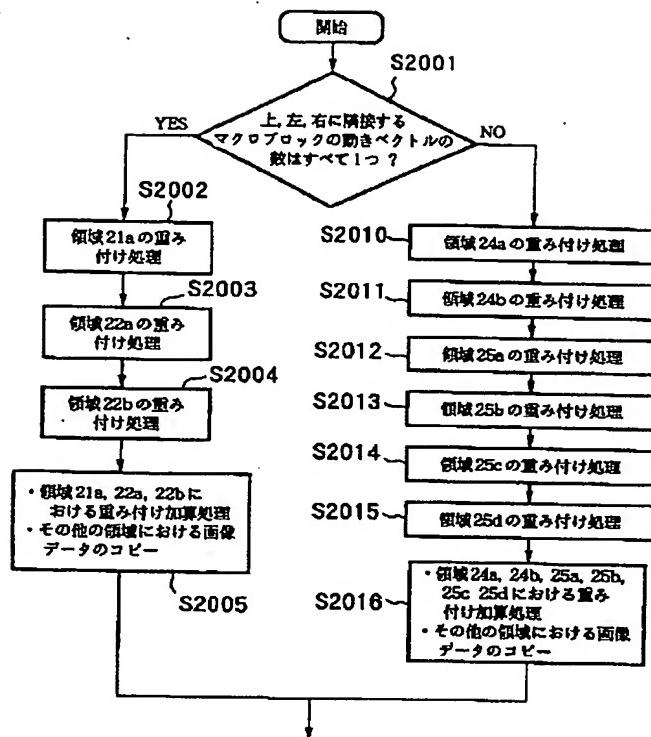
〔图 19〕



[図20]

〔図22〕

[图 14]



【図21】

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-299864  
(43)Date of publication of application : 24.10.2000

(51)Int.CI.

H04N 7/32

(21)Application number : 11-104088

(71)Applicant : CANON INC

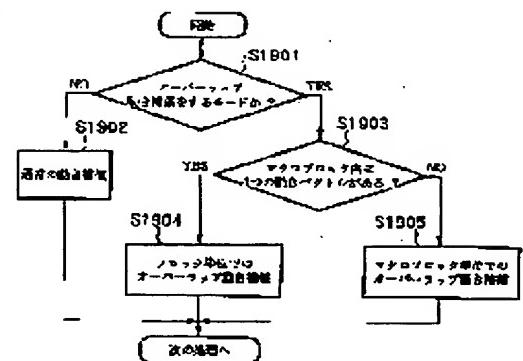
(22)Date of filing : 12.04.1999

(72)Inventor : NAKAYAMA TADAYOSHI

## (54) METHOD FOR PROCESSING DYNAMIC IMAGE

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To fast perform overlap motion compensation that reduces discontinuous distortions between adjacent blocks caused by motion vector prediction by generating a weight coefficient matrix corresponding to a macroblock and performing overlap motion compensation prediction coding in a macroblock unit by using it.  
**SOLUTION:** It is identified whether or not a mode for performing overlap motion compensation is (S1901), and when it is not the mode, normal motion compensation is executed (S1902). Meanwhile, in the case four motion vectors exist in a macroblock when it is decided as a mode performing overlap motion compensation, overlap motion compensation processing in a general block unit is performed (S1904). In the case not four motion blocks but only one block exists, motion compensation processing is performed in a macroblock unit (S1905). Thus, it is possible to reduce the number of pixels to be accessed and the number of add operations.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

**2·2 an English language explanation**  
of Japanese laid-open patent publication No. 2000-299864

This document describes the fast processing method for overlapped block motion compensation.